Unit 7: Using External Data Sources

Generate something distinctive and new—a mashup of sites.

Site:	School of Computing and Information Systems
Course:	Computer Science 266: Introduction to Web Programming (Revision 3)
Book:	Unit 7: Using External Data Sources
Printed by:	Hector Barquero
Date:	Thursday, 19 September 2024, 11:57 AM MDT

Table of contents

Study Guide Unit 7 Using External Data Sources

Overview

☞ Unit 7 is worth 10% of your portfolio grade. **When you have completed Unit 7, you will submit your portfolio for final grading.** See the **Final Submission of Work** section of Unit 0: Orientation.

Podcast - Introduction to Unit 7

This unit requires you to find one or more services that are available on the Web and incorporate data and/or functionality from those services into your site. This is not about just embedding parts of other sites in your own: it's about receiving and sending data to and from them that you use to generate something distinctive and new—a mashup of sites. You may have already discovered some of this while looking at the JQuery libraries, several of which are concerned with using AJAX. In this exercise you will be directly manipulating and using APIs offered by external websites such as Twitter, Google, Yahoo, Foursquare, or Facebook in order to enhance the functionality of your site in accordance with your previously stated goals and purposes.

Study Guide Unit 7 Using External Data Sources

Learning Outcomes

When you have completed this unit, you should be able to use JavaScript to access and use web services for dynamic content (AJAX, JSON, etc.).

Study Guide Unit 7 Using External Data Sources

Problem

Important: do not begin this task until you have completed all the previous units, including your reflective learning diary for each unit.

Find appropriate services, gadgets, widgets, and so on to use in your site based on the personas, scenarios, and other design decisions made for Unit 1. Google is a good source of these, or Yahoo, or Microsoft. Add those you choose to your site, ensuring that the look and feel is in keeping with the overall look and feel of your site. Examples of ways you might include external services include (but are by no means limited to)

- a form that is auto-filled from a database as the user types.
- a widget that enables people on your site to communicate with each other in real-time.
- a widget that gives access to your Facebook status.
- an embedded map that shows the locations of other users of the site.
- an embedded Google Gadget that allows people to comment on pages or articles on your site.
- a dynamic list of related sites drawn from del.icio.us.
- a button to recommend this site on different services (e.g. ,Facebook, LinkedIn, Orkut).
- latest Twitter posts relating to the site's themes and purposes (maybe via hashtags).

You are encouraged to use more than one service to enhance the experience of the user; however, your use of external sites should be driven by identified needs rather than technology so "enough" in this case is what suits your themes, purposes, scenarios, and personas.

We would expect you to use **at least two** services and at most ten, but we will take into account the amount of processing that you add using JavaScript and not just count the number of services used, as well as taking the purposes, etc., of the site into account. It is possible that a single service, if used in an interesting and relevant enough way, might be sufficient to meet the outcomes, but normally more would be needed.

Marks will be given for effective use of technology as well as the degree to which the service enhances the user's experience of your site, bearing in mind the scenarios and personas identified and built upon since Unit 1.

Study Guide Unit 7 Using External Data Sources

Process Guide

Consider what kinds of facilities you would like to provide on your site, going back to the themes, purposes, scenarios, and personas of Unit 1 for inspiration.

Investigate and explore sites offering an API, browsing examples and documentation, bearing in mind at all times what you wish your site to achieve.

Take a few tutorials on APIs that interest you. There is a list below, but most sites that offer APIs also offer tutorials on using them. Rest assured, once you have figured out the jargon and buzzwords like JSON and AJAX, it is generally a pretty easy thing to do for someone who has reached your level of proficiency. In essence, it usually involves sending and receiving data from external sites in a straightforward format, then doing something with those data at your end using JavaScript.

Be aware that some tutorials and methods require server-side processing: avoid those unless you have access to that server and are proficient in programming it to work as you wish!

Decide upon one or more sites that you would like to use to enhance the user experience on your site. Write a brief proposal explaining what you wish to do and submit this to your tutor. Your tutor may approve this or suggest alternatives and modifications that will better suit your needs for a good site and good marks.

Check with your tutor that you are heading along the right lines and wait for feedback before completing this exercise. Use the 'Proposal for external data use review' link on this Moodle site to submit your proposal to your tutor.

Integrate the technology of your choice with your site.

Upload your changes to your SCIS website.

As always, don’t forget to **write your reflections** on the process, challenges, and things you found about yourself in your reflective learning diary.

Important: create a zipped-up version of the code (no need for the images) and save it as an attachment to your diary entry, as you should have done at the end of each unit. This will help to provide us with a clearer image of how your site has developed over the course, what you have learned and how you have developed your skills.

Study Guide Unit 7 Using External Data Sources

(Almost) Everything You Need to Know about Using External Site Data in a Few Short Paragraphs

You will mostly be using AJAX to access external site data. You may additionally use other techniques for incorporating external site data such as iFrames if you wish—this may be useful in fitting your site with your scenario, but remember that the main purposes and marks for this assignment are, of course, JavaScript-oriented.

AJAX stands for Asynchronous JavaScript and XML, a term coined by Jess James Garrett to describe a trend in web technology rather than one particular method or tool. In fact, it's a slight misnomer because, although all related approaches do use JavaScript, there are other common ways to send and grab data from other sites and to use it on your page that do not use XML, notably JSON, but also plain HTML.

It is not a specific set of functions and technologies, more a set of methods that allow you to update pages (technically, parts of the DOM) without reloading the whole page using data from a source that is not the page itself, and to send data from a page to a receiving application without having to leave the current page.

You may have already come across AJAX in previous units, especially when using JQuery (which has rich AJAX libraries), but in this exercise we want you to write the code yourself (though you are welcome and encouraged to use properly ascribed code snippets by others, especially sites offering APIs, to do some of the work for you).

Study Guide Unit 7 Using External Data Sources

XMLHttpRequest

The most important JavaScript object that you will need to know about is XMLHttpRequest. It can be used to both send and retrieve XML (as the name suggests) including (of course) XHTML, but is also quite capable of reading in non-XML variants of HTML, plain text, and certain useful standards including JSON (JavaScript Object Notation).

JSON is a standard format for working with JavaScript objects that makes it very easy indeed to make use of objects from other sites in your scripts—for example, to read in data provided with APIs by Google Maps, Twitter, Facebook, or Gmail.

XMLHttpRequest is also a really useful means of dynamically changing parts of your page using static pages or XML data on your own site. It is the way that Google Search offers real-time suggestions as you type, how forms-based sites bring up suggestions about what you are typing, how mashup sites bring data from maps and photos together, and plenty more.

Note that older Microsoft browsers deal with this a little differently than all modern browsers and require extra effort to code for: to cater to as many potential audiences as possible, you may need to bear this in mind (you might find JQuery's browser detection facilities useful here).

Study Guide Unit 7 Using External Data Sources

Using APIs

An API (application programming interface) is a hook provided by writers of a program to let other programs make use of some of its functionality. Typically it will take the form of an object, function, or similar to which you can send or receive data. APIs often provide methods to control some of their functions and/or properties that you can query.

In many cases, use of an API will be restricted to avoid abuse. You will often find that you have to sign up for a service in order to receive a unique key that will give access to the system you are trying to talk to. If you already have an account with the provider (e.g., Google, Facebook, Yahoo, Twitter), then this is unlikely to be more than a minor technical problem. However, please use care and judgement when giving away personal information if you need to sign up for a service.

This is important: Always read agreements and policies regarding privacy, ownerships, terms of service, licence to use the service, and related documentation, and do not proceed if you have any concerns or doubts about the safety of your personal information or that of others. If in doubt, talk with your tutor via the discussion forum on the Landing. Remember that there are many services that you might use for this exercise so, if you are not happy with the conditions or protection required by a service, you should not feel obliged to use it: just choose another.

Canadian, European, and some other sites are well protected by privacy and data protection legislation, but those in some other countries (notably the US) do not, at the time of writing, provide such firm assurances in law. Unfortunately, many of the more interesting sites (Google, Facebook, Twitter, Yahoo, for instance) are at least partially hosted in the US. We do not wish to overemphasize the risks, which, for the most part, are negligible, but you should always be aware that caution must be exercised, and be alert when parting with any of your personal information.

Study Guide Unit 7 Using External Data Sources

Using a Web Server, Not the File System, to Host Your Files

Note that because it explicitly uses HTTP and not file system access, XMLHttpRequest *cannot* be used to access file content on a local machine *unless* you are running a web server. When you open a local file, your browser is not sending any HTTP headers and will not receive any HTTP responses.

What this means is that if you simply try to open a web page using XMLHttpRequest from your file manager, it won't be able to read other local content because you will be using the file protocol and not the http protocol. This is an easy thing to spot because that is what will prefix the URL in your browser: file:/// or http://. To test your code properly using the HTTP protocol, you can either put it on your site at Athabasca (or some other web server if you have access to one) or set up a local web server on your machine.

If your computer doesn't already have a web server installed (all Mac and Linux machines will probably have one by default, as will most Windows versions apart from the Home editions, though you may need to enable and configure the machine to use it effectively), see links for this unit and/or on the Landing bookmarks for the COMP 266 group for some straightforward-to-install servers.

Study Guide Unit 7 Using External Data Sources

JavaScript Security Restrictions

Remember that JavaScript is a restricted language that, significantly, does not allow you to read or manipulate pages or data from a site other than the one on which the script is hosted. This is a security precaution to prevent cross-site scripting attacks and is a good thing under most circumstances, but can be a nuisance if you need to do it for legitimate purposes.

There are various ways to get around this restriction, including using a server-side system like PHP, Ruby, ASPX, or Perl on your own site to get the data for you. However, it is possible to take advantage of some of the things that the security models of browsers *will* allow, in particular using JSONP (JSON with padding). See http://en.wikipedia.org/wiki/JSONP#JSONP for more information on this, as well as other tutorial sites that go into more detail. You will also find plenty of examples in the documentation for different sites offering APIs and associated tutorials on the Web of how to circumvent the restrictions.

Most sites that offer APIs are keen to let you make use of their data and will have provided means by which you can do so. Read the documentation carefully and follow it. Your tutor may be able to help, but remember that he or she is unlikely to be familiar with every site or way of using a site that might be selected here so don't expect step-by-step solutions: your tutor will try to help with approaches to problem solving but may not know the exact answer in advance. However, your class-mates might! Do make full use of the discussion forum and shared bookmarks on the Landing in seeking help.

☞ **When you have completed Unit 7, you will submit your portfolio for final grading.** See the **Final Submission of Work** section of Unit 0: Orientation.

Study Guide Unit 6 Using Libraries

Indicative Grading Criteria

Grade	Criteria
A	• Well chosen and broad range of external sites that greatly improve the experience for your visitors, based very clearly on the themes, purposes, scenarios, and personas developed for Unit 1. • Code integrated effectively with your site with a significant amount of post-processing once the data are received. • More than one external service used, probably mashing up content and data from multiple sites (e.g., Twitter login and Google maps). • Careful attention to usability and accessibility: the site will continue to work or fail gracefully whether or not a person has an identity on services added or if the service becomes unavailable • Elegant, maintainable code (well written, nicely laid out, well-commented; good separation of data, processing, and presentation).
B	• Two or three well-chosen external sites that improve the experience for your visitors, based on the themes, purposes, scenarios, and personas developed for Unit 1. • Code integrated effectively with your site with some post-processing of the data received. • More than one external service used, probably amalgamating or aggregating content and data from multiple sites (e.g., Twitter login and Google maps). • Careful attention to usability and accessibility: the site will continue to work or fail gracefully whether or not a person has an identity on services added or if the service becomes unavailable. • Maintainable, well-written, well-commented code.
C	• A couple of external site services that improve the experience for your visitors, with an identified and relevant connection to the themes, purposes, scenarios, and personas developed for Unit 1. • Code mostly integrated effectively with your site, with some post-processing involved. • Some integration/aggregation between content from other sites. • Usability and accessibility mostly effective. • The site may not always fail gracefully if a user does not have access to services from the external site or sites. • Code not hard to maintain, but little attention to code design.

D

- A couple of external site services that marginally improve the experience for your visitors, with some connection to the themes, purposes, scenarios, and personas developed for Unit 1.
- Code mostly integrated effectively with your site.
- Limited range of integration between content from other sites.
- Limited usability and accessibility: the site may not fail gracefully if a user does not have access to services from the external site or sites.
- Code difficult to maintain.

Study Guide Unit 7 Using External Data Sources

Resources

☞ If you find other useful resources, please add links to them on the Landing so that others may benefit. You may use such activities as supporting evidence of having met the learning outcomes.

Sites Offering Services and APIs

Yahoo tools and APIs: http://developer.yahoo.com/everything.html (includes widgets and other goodies)

Facebook: http://developers.facebook.com/docs/guides/web

Twitter: http://apiwiki.twitter.com/w/page/22554648/FrontPage

Google Maps: http://code.google.com/apis/maps/index.html

Google APIs: http://code.google.com/more/ (also see http://code.google.com/ for a very wide range of tools and gadgets)

MySpace APIs etc: http://developer.myspace.com/wordpress/

YouTube player API: http://code.google.com/apis/youtube/js_api_reference.html

Flickr API: http://www.flickr.com/services/api/

Various Tutorials

AJAX tutorial from W3Schools: http://www.w3schools.com/ajax/default.asp

3-minute JSON tutorial: http://secretgeek.net/json_3mins.asp

Introduction to JSON: http://www.ajaxprojects.com/ajax/tutorialdetails.php?itemid=11

Using the Twitter API: http://www.webmonkey.com/2010/02/get_started_with_the_twitter_api/

W3C specification: http://www.w3.org/TR/XMLHttpRequest/

Web Servers for Local Installation

XAMPP: http://www.apachefriends.org/en/xampp.html

XAMPP is a nice, free, simple-to-install distribution of Apache, MySQL, PHP, Perl, and much more besides that is available for Windows, Linux, MacOS, and Solaris. Highly recommended.

It should work for most people by simply unzipping and running it, but if you run into problems, the chances are that it might be clashing with other servers already installed on your machine that are using the same TCP/IP port addresses. Either turn these off or use different port addresses to avoid a clash. Documentation on such issues is available on the site.